Interaction Graphs: Full Linear Logic

Thomas Seiller

Department of Computer Science, University of Copenhagen, Njalsgade 128, bygning 24, 2300 København S, Denmark
seiller@di.ku.dk

Abstract

Interaction graphs were introduced as a general, uniform, construction of dynamic models of linear logic, encompassing all *Geometry of Interaction* (GoI) constructions introduced so far. This series of work was inspired from Girard's hyperfinite GoI, and develops a quantitative approach that should be understood as a dynamic version of weighted relational models. Until now, the interaction graphs framework has been shown to deal with exponentials for the constrained system ELL (Elementary Linear Logic) while keeping its quantitative aspect. Adapting older constructions by Girard, one can clearly define "full" exponentials, but at the cost of these quantitative features. We show here that allowing interpretations of proofs to use continuous (yet finite in a measure-theoretic sense) sets of states, as opposed to earlier Interaction Graphs constructions were these sets of states were discrete (and finite), provides a model for full linear logic with second order quantification.

Categories and Subject Descriptors F.3.2 [*Semantics of Programming Languages*]: Denotational Semantics

General Terms Semantics; Quantitative Models

Keywords Interaction Graphs; Linear Logic; Geometry of Interaction; Quantitative Semantics; Measurable Dynamics

1. Introduction

This work deals with so-called dynamic models of proof theory, such as game semantics and geometry of interaction. It extends previous work providing a uniform construction of quantitative dynamic models of (fragments of) linear logic to full linear logic with second-order quantification.

Geometry of Interaction. A Geometry of Interaction (GoI) construction, i.e. a construction that fulfills the GoI research program [17], is in a first approximation a representation of linear logic proofs that accounts for the dynamics of cut-elimination. Contrarily to denotational semantics, a proof π and its normalised form π' are not represented by the same object, but they remain related through a semantic interpretation of the cut-elimination called the *execution* Ex: $\mathrm{Ex}(\pi) = \pi'$. A GoI construction hence represents both the proofs and their normalisation; it is in some ways an untyped variant of game semantics [22].

The further aim of geometry of interaction is to reconstruct logical operations from such a dynamic representation of proofs. The objects of study in a GoI construction are actually a generalisation of the notion of proof – sometimes called paraproofs. This point of view allows a reconstruction of logic as a description of how these objects interact in the same spirit as realisability [24–26]: a program is of type $\mathtt{nat} \to \mathtt{nat}$ because it produces a natural number when given a natural number as an argument. As in game semantics and classical realisability, one can however describe a necessary condition for being the interpretation of a proof, and defines *winning paraproofs* as those objects satisfying it.

In spite of their seemingly deep abstraction, the GoI constructions provide mathematical models which are very close to actual computing. As an illustration of this fact, let us mention the *Geometry of Synthesis* program initiated by Ghica [9–11, 13]. This research program, inspired by geometry of interaction, aims at obtaining logical synthesis methods for VLSI designs.

Quantitative Semantics. Quantitative semantics find its origins in Girard's work on functor models for lambda-calculus [15]. This work, which predates its seminal work on linear logic [14] and actually inspired it, exhibits for the first time a decomposition of the semantic interpretation of lambda-terms as Taylor series. These series capture a number of information about the time, space, resource consumption of the programs it represents. Quantitative semantics are therefore more involved than so-called *qualitative* semantics, since they mirror more information about the programs that are interpreted. Recently, quantitative semantics has been used to give denotational semantics for various algebraic extensions of lambda calculus such as probabilistic [3] or differential lambda calculi [7]. Work by Laird, Manzonetto, McCusker and Pagani on weighted relational models [29] provides a uniform account of several denotational models accounting for quantitative notions, using a refinement of the relational model.

Interaction Graphs. Interaction graphs were first introduced by the author [33] as a combinatorial approach to Girard's hyperfinite Geometry of Interaction [20], restricted to the multiplicative fragment of linear logic. An extension capable to deal with additive connectives was then defined [37] and shown to abstract not only the (additive fragment of the) hyperfinite GoI model but all previously introduced GoI constructions as well. Both papers proposed a model construction in the spirit of Girard's GoI construction where proofs were interpreted by graphs instead of infinite operators. Dealing with exponentials however needs one to consider infinite objects. This is why a third paper [38] showed how the construction on graphs can be applied when working with a generalisation of graphs named *graphings*. Graphings are in some sense *geometric realisations* of graphs on a measure space $\mathbf{X}$ which were first introduced in the context of ergodic theory [1, 8]. This allows not only to consider infinite graphs (which can be used to define exponentials in the same way as the original GoI constructions), but also graphs acting on continuous, thus infinite but finite-measure

spaces. This general construction on graphings was shown [38] to improve on Girard's hyperfinite GoI [20] since it allows a satisfactory treatment of second-order quantification. Lastly, a fourth paper[1] [35] showed how the consideration of graphings can be used to define "quantitative" exponential connectives for Elementary Linear Logic [18], a fragment of linear logic that captures elementary time computation [4].

Unbounded Exponentials and Quantitative Aspects. The author's work on Interaction Graphs should be understood [36] as a *dynamic counterpart* of weighed relational models by Laird *et al.* [29], i.e. its relation to standard dynamic models (geometry of interaction, game semantics) is comparable to weighted relational models' relations with standard relational models. Indeed, it provides a uniform construction of models which not only captures all of Girard's GoI models, but also extends them: while Girard's constructions can be understood as interpreting proofs as graphs[2], we here interpret proofs as *weighted graphs*, i.e. graphs with weighted edges[3]. Furthermore, interaction graphs models can reflect these quantitative information at the level of types since the latter are built from an *orthogonality relation* which can take those weights into account. Indeed, the orthogonality relation is defined through a measurement of cycles [38] by means of an integral over a finite-measure space – the *support* of the cycle. In the simplest cases one measures a cycle π of suppport $\mathrm{supp}(\pi)$ and weight $\omega(\pi) \in \Omega$, along a measurable map $m : \Omega \to \mathbf{R}_+$, by the following integral:

$$\int_{\mathrm{supp}(\pi)} m(\omega(\pi)) \tag{1}$$

Since Interaction Graphs provide a generalisation of Girard's constructions, one could easily adapt the interpretation of exponential connectives from Girard's first constructions [16, 19] to obtain a model of full linear logic. This adaptation would extend to Danos' interpretation of pure lambda-calculus in GoI [2]. However, this interpretation of exponential connectives corresponds to defining $!a$ as a (countable) infinite family of copies of a. Thus, even if a is represented by a graphing acting on a space of finite measure, its exponentiated version $!a$ acts on a space of infinite measure. This fact hinders the quantitative aspects of our model since it creates cycles π whose support $\mathrm{supp}(\pi)$ are spaces of infinite measure. As a consequence, the integral defining the orthogonality relation (Definition 1) diverges as soon as the weight is not mapped to 0, i.e. as soon as $m(\omega(\pi)) \neq 0$. The resulting model is therefore no longer capable of depicting quantitative information.

Contributions. We define, in the framework of interaction graphs, exponential connectives for full linear logic in a way that preserves the quantitative aspects of the construction, providing the first dynamic model of second-order linear logic accounting for quantitative aspects. Indeed, to the author's knowledge, there exists no game semantics or GoI models for this logical system which include quantitative features. Indeed, although Girard's so-called GoI3 construction [19] provides a model of this same logical system, the treatment of exponential connectives in the latter work prevents from any generalisation accounting for quantitative information, as already explained. On the side of game semantics, the quantitative game semantics for linear logic of dal Lago and Laurent [28] does not deal with additives and quantifiers and moreover seems more limited than our own models in the range of quantitative features it can accommodate.

Moreover, we are able to pinpoint the computational principles (represented as measurable maps) that are essential to interpret digging and dereliction, providing new insights on constraint linear logic and their semantics. Formally, this is obtained by exhibiting a single map – the *exchange* `xch` – which turns a model of ELL into a model of LL. Indeed, as discussed after Definition 9, removing the exchange restricts our model to the model of ELL based on graphings [34, 35].

Outline of the paper. We define exponential connectives along the same lines as in our work on ELL (thus bounded) exponentials [35], avoiding the involvement of infinite-measure sets. With this definition of exponential connectives, one would however expect only a restriction of linear logic, such as ELL. To bypass this restriction, we relax the notion of states. Indeed, the interpretation of proofs in interaction graphs makes use of so-called *thick graphs* – or *thick graphings* in the general framework –, which can be understood as graphs with states. While previous work considered only finite sets of states, we loosen this definition to allow for infinite yet finite-measure (actually continuous) sets of states. This modification impacts slightly on the basic notions notions and constructions considered in previous work [38], for which we introduce adequate generalisations. These changes, however, do not raise any technical difficulties. The resulting model is then shown to model digging and dereliction in addition to the principles of Elementary Linear Logic, thus interpreting full linear logic. Finally, we discuss the issue of the representation of cut-elimination in the model.

2. Interaction Graphs

We start by a discussion meant to give intuitions about the basic principles at work in the interaction graphs models. We illustrate those principles by explaining the notion of *thick and sliced graphs* [35]. This discussion is quite informal in that we will only provide explicit and complete definitions of the objects and operations that are essential for the understanding of this paper, to avoid overloading the reader with non-essential definitions. Indeed, the actual model uses thick and sliced *graphings*, a generalisation needed to accomodate both exponentials and quantifiers. Before providing the formal definition of those at the end of the section, we discuss the notion of "graphs with states" and how it can be generalised to continuous sets of states.

2.1 Thick and Sliced Graphs

The term "graph" will stand for "directed weighted graphs", i.e. directed graphs with a weight function from the set of edges to a monoid[4] of weights Ω. Given a graph G, we will always denote E^G its set of edges, V^G its set of vertices, t^G and s^G its target and source maps, and ω^G its weight map.

The notion of *thick* graphs generalises that of graphs by introducing a set of (control) states – called a dialect. A graph G with dialect D^G is nothing more than a graph whose set of vertices is of the form $V^G = S^G \times D^G$ – the set S^G is called its *support*. The set D^G then acts as a set of control states when considering the operation of *execution*, which represents the cut-elimination procedure. When working with (non-thick) graphs, this operation is represented as the computation of a graph of alternating paths; the notion of alternating path between thick graphs gives a particular role to the dialects. Indeed, an alternating path between thick graphs G_0

[1] Although the author's PhD thesis [34] did not contain the general treatment of graphings [38], it already introduced the model of ELL [35] and the restricted theory of graphings this model uses.

[2] Girard interprets proofs as partial isometries acting on a Hilbert space $\mathbb{H}$ which, by considering the right basis for $\mathbb{H}$ correspond to graphs.

[3] Actually, the most general models are built around the lesser known notion of weighted *graphing*. However, thinking about graphings as graphs should provide the reader with the right intuitions.

[4] As we consider paths in the following, the structure of monoid is essential as it allows to define the weight of a path as the product of the weights of the edges that it is composed of.

and G_1, with respective dialects D^{G_0} and D^{G_1}, is a finite sequence of edges $e_0 e_1 \dots e_k$ and a sequence of triples $(s_i, g_i^{(0)}, g_i^{(1)})_{i=0}^{k+1}$ such that:

(Alternation) $e_i \in E^G$ if and only if $e_{i+1} \in E^H$;

(States) if $e_i \in E^{G_j}$ then $\begin{cases} s^{G_j}(e_i) &= (s_i, g_i^{(j)}) \\ t^{G_j}(e_i) &= (s_{i+1}, g_{i+1}^{(j)}) \\ g_i^{(1-j)} &= g_{i+1}^{(1-j)} \end{cases}$;

The interpretation of the dialect as a set of (control) states comes from the way its elements are dealt with in the above definition. Now, given two thick graphs G, H, the intersection of their supports represent a *cut*; the result of the elimination of this cut is called the *execution* of G and H. It is defined as the thick graph $G :: H$, of support the symmetric difference $S^G \Delta S^H$, of dialect the product $D^G \times D^H$, whose edges are exactly the alternating paths between G and H whose source and target lie outside of the cut. This is reminiscent of game semantics' *composition and hiding*: composition corresponds here to the computation of all alternating paths, while hiding corresponds to the restriction to those paths starting and ending outside the cut. The formal definition of execution, in the framework of graphings, is given at the end of this section.

Now, *thick and sliced graphs* further extend this notion of thick graphs by considering finite formal weighted sums $\sum_{i \in I^G} \alpha_i^G G_i$ where I^G is a finite indexing set, the coefficients α_i^G are real numbers and $\{G_i\}$ is a set of thick graphs sharing the same support (but not sharing the dialects). This notion is crucial for treating additive connectives [37]. The operation of execution is then extended "by linearity" (although the sums are not linear combinations), letting:

$$\left(\sum_{i \in I^G} \alpha_i^G G_i\right) :: \left(\sum_{i \in I^H} \alpha_i^H H_i\right) = \sum_{(i,j) \in I^G \times I^H} \alpha_i^G \alpha_j^H G_i :: H_j$$

2.2 Continuous Dialects

Interaction graphs models dealing with exponential connectives of linear logic are based on the notion of *thick and sliced graphings*, obtained by a second layer of generalisation over this notion of thick and sliced graphs [34]. While graphings will be introduced formally in the next section (Definition 2), we provide an intuitive description to discuss this generalisation. This discussion can be skipped in a first read, as only the formal definition of microcosm is needed to follow the next section.

Graphings are in some sense *geometric realisations* of graphs on a measure space $\mathbf{X}$. Specifically, a graphing G is defined as a graph such that for each edge $e \in E^G$, $s^G(e)$ and $t^G(e)$ are measurable subsets of $\mathbf{X}$, and there is a measurable map $\phi_e^G : s^G(e) \to t^G(e)$ which *realises* e. As for graphs, one can define *thick and sliced graphings* by first defining thick graphings – graphings with a dialect, then consider formal weighted sums of those. It is natural, while working with graphings, to consider dialects themselves as measure spaces, and more precisely (finite) discrete probability spaces. A thick graphing of dialect $\mathbf{D}$ is then easily described as a graphing over the measure space $\mathbf{X} \times \mathbf{D}$.

The purpose of the current work is to extend this definition to allow for *continuous dialects*, i.e. continuous measure spaces in place of discrete ones. We will show how to define in this setting the interpretation of second order linear logic without hindering the "quantitative" features of the interaction graphs construction. This however comes with a small drawback in the form of a minor complexification of the framework, which we now explain.

We did not dwell on this point earlier, but thick graphs (graphs with dialects) are considered *up to*[5] renaming of their dialect; a thick graph G which is a dialect-renaming of a thick graph F is called a *variant* of F (Definition 3). To define correctly this notion of variant one needs to consider bijections between the dialects. However, when considering graphings and replacing the dialects with possibly continuous probability spaces, we face a problem when considering the following two probability spaces: $\mathbf{k} = \{1, \dots, k\}$ with discrete measure, and $[\mathbf{0}, \mathbf{1}]$ with Lebesgue measure. Indeed, any thick graphing G with dialect $\mathbf{k}$ has a variant H with dialect $[\mathbf{0}, \mathbf{1}]$: each element $i \in \mathbf{k}$ is represented by the interval $I_i = [i/(k+1), (i+1)/(k+1)]$, and an edge of source (v, i) and target (v', j) realised by a map $\phi : (v, i) \to (v', j)$ in G is realised in H by $\phi_1 \times T_{i,j}$ where $T_{i,j}$ is the translation $x \mapsto x + (j - i)/(k + 1)$ and ϕ_1 is the map $v \to v'$ underlying[6] ϕ. However, this cannot be formalised through an adequate notion of bijection: here we would expect *Borel isomorphisms* since we work with measure spaces, but no such isomorphism exists between $\mathbf{k}$ and $[\mathbf{0}, \mathbf{1}]$. To avoid these troubles, we will therefore consider all our dialects to be isomorphic to $[\mathbf{0}, \mathbf{1}]$ with its Lebesgue measure. Since, as we just explained, a graphing with discrete dialect always has a "variant" with $[\mathbf{0}, \mathbf{1}]$ as dialect, and since thick graphings are considered *up to* renaming, this restriction is seamless.

The second change from earlier work [38] is that we need to consider an extension of the notion of *microcosm*. A microcosm $\mathfrak{m}$ was defined as a monoid of measurable maps $\mathbf{X} \to \mathbf{X}$ used to consider "restrictions" of the model to $\mathfrak{m}$-graphings: graphings whose realisers – i.e. the maps that realises edges – are restrictions of maps in $\mathfrak{m}$. This original notion of microcosm did not incorporate the dialect. This is explained by the fact that the latter was discrete, and therefore any measurable maps realising an edge in a thick graphing could be described as a product of a measurable maps from $\mathbf{X}$ to $\mathbf{X}$ with a partial bijection on the dialect. Now that we allow for continuous dialects, one can consider realisers of edges that do not simply arise in this way from[7] a map $\mathbf{X} \to \mathbf{X}$. The following definition therefore adapts (in fact extends) the previously considered notion of microcosm in a very natural way in order to incorporate this change. Let us stress that for technical reasons discussed in earlier work [38], the measurable maps considered should be non-singular transformations[8] which are measurable-preserving, i.e. map measurable sets to measurable sets.

Definition 1 (Microcosm)**.** Let $\mathbf{X}$ be a measure space. A *microcosm* is a monoid (for the composition of functions) of measurable-preserving non-singular transformations $\mathbf{X} \times [\mathbf{0}, \mathbf{1}] \to \mathbf{X} \times [\mathbf{0}, \mathbf{1}]$.

2.3 Graphings and Exponential-Free Linear Logic

This section is meant to recall the main results of previous work [38], to which we refer the reader for a complete picture. We first define weighted (thick) graphings, a generalisation of the homonymous notion considered by Adams [1] and later by Gaboriau [8].

Definition 2 (Graphing)**.** Let $\mathfrak{m}$ be a microcosm, Ω a monoid of weights, S^G a measurable subset of $\mathbf{X}$ and $\mathbf{D}^G$ a probability space isomorphic to $[\mathbf{0}, \mathbf{1}]$. A thick Ω-weighted $\mathfrak{m}$-graphing G of support S^G and dialect $\mathbf{D}^G$ is given by a set of *edges* E^G and $\forall e \in E^G$:

[5] It is important to remark here that we don't consider the set of graphings quotiented modulo renaming, but we want to be able to formalise this notion of equivalence.

[6] Since $\mathbf{k}$ is discrete, any measurable map $\phi : (v, i) \to (v, j)$ is defined from a measurable map $\phi_1 : v \to v'$ by $\phi(x, i) = (\phi_1(x), j)$.

[7] As an example, one can consider the *exchange* map defined below (Definition 9) and which is needed to interpret both digging and dereliction.

[8] Let $\mathbf{X} = (X, \mathcal{B}, \mu)$ be a measure space. A measurable map $f : X \to X$ is non-singular when $\forall A \in \mathcal{B}, \mu(f^{-1}(A)) = 0 \Leftrightarrow \mu(A) = 0$.

- a *source* $s^G(e)$, i.e. a measurable subset of $S^G \times D^G$;
- a *realiser* $\phi_e^G \in \mathfrak{m}$ such that $\phi^G(s^G(e)) \subset S^G \times D^G$;
- a *weight* $\omega^G(e)$.

For all edge $e \in E^G$, one can then define the *target* $t^G(e)$ of e as the measurable subset $\phi_e^G(s^G(e))$.

A graphing G is *dialect-free* if it does not make use of its dialect, i.e. if for all edge e, $\phi_e^G = \tilde{\phi}_e^G \times \mathrm{Id}_{\mathbf{D}^G}$, with $\tilde{\phi}_e^G : \mathbf{X} \to \mathbf{X}$.

Notations. Let A be a graphing, B a Borel automorphism of $\mathbf{X} \times [\mathbf{0},\mathbf{1}]$. We denote $B(A)$ the graphing whose edges are $B^{-1} \circ \phi \circ B$; up to the automorphism between $\mathbf{D}^A$ and $[\mathbf{0},\mathbf{1}]$. When B is a Borel automorphism of $\mathbf{X}$, we abusively denote by $B(A)$ the graphing $B \times \mathrm{Id}_{[\mathbf{0},\mathbf{1}]}(A)$. We also denote by $A \times \mathrm{Id}_{[\mathbf{0},\mathbf{1}]}$ the graphing of dialect $\mathbf{D}^A \times [\mathbf{0},\mathbf{1}]$ whose edges are realised as $\phi_e \times \mathrm{Id}_{[\mathbf{0},\mathbf{1}]}$.

Definition 3 (Variants)**.** Let F and G be graphings. If there exists a Borel automorphism $\phi : [\mathbf{0},\mathbf{1}] \to [\mathbf{0},\mathbf{1}]$ such that $F = \mathrm{Id}_{\mathbf{X}} \times \phi(G)$, we say that F and G are variants.

Morally, graphings are sort of graphs which offer richer combinatorics since two vertices might have a non-trivial intersection without being equal. In particular, when considering paths, one should be careful about the sources and targets: a path in a graphing G is a sequence of edges $\pi = e_1, e_2, \dots, e_k$ in E^G such that not only $s^G(e_{i+1}) \cap t^G(e_i)$ is non-negligible for every i, but also verifying that every sequence

$$(\phi_{e_i}^G \circ \phi_{e_{i-1}}^G \circ \dots \phi_{e_1}^G)(s^G(e_1))$$

is of strictly positive measure. This path is then naturally realised as the composite $\phi_\pi^G = \phi_{e_k}^G \circ \dots \circ \phi_{e_1}^G$, and is considered with its *maximal domain* $s^G(\pi)$, i.e. the set of all x such that for all i, $\phi_{e_i}^G \circ \dots \circ \phi_{e_1}^G(x) \in s^G(e_{i+1})$, and its codomain $t^G(\pi) = \phi_\pi^G(s^G(\pi))$. The weight of π is obviously defined as $\omega^G(\pi) = \omega^G(e_k)\omega^G(e_{k-1})\dots\omega^G(e_1)$ using the composition law of Ω.

We can then define *alternating path* between thick graphings as in the case of graphs, and introduce the operation of *execution* between thick graphings, the semantic counterpart to the cut-elimination procedure. We write $\mathrm{AltPath}(F,G)$ the set of all alternating paths between two graphings F, G.

Before defining execution, we need to introduce an additional construction on paths that will allow us to restrict them to a subset of their domain, i.e. perform the "hiding part" of game semantics' composition. Given a path π in a graphing G and a measurable subset C (thought of as the cut), we define ${}^C[\pi]_o^o$ as the path with same realiser and weight as π, and whose source has been restricted to the measurable set $s^G(\pi) \cap \bar{C} \cap (\phi_\pi^G)^{-1}(\bar{C})$, where $\bar{C}$ is the complement set of C. Intuitively, we restrict π to the maximal subset of its domain that lies outside of C and whose image through the realiser ϕ_π^G lies outside of C.

Definition 4 (Execution)**.** Let F and G be graphings with $S^F = V \uplus C$ and $S^G = C \uplus W$. Their *execution* $F :: G$ is the graphing of support $V \uplus W$ defined as the set of all restrictions ${}^C[\pi]_o^o$ for alternating paths $\pi \in \mathrm{AltPath}(F,G)$.

Example 1. We consider the two one-edge graphings (without dialects or weights to be concise) G and H illustrated on the left-hand side of Figure 1. The edge of G has source the segment $[0,2]$, target the segment $[4,6]$ and is realised by the map $x \mapsto 6-x$. The edge of H has source the segment $[5,6]$, target the segment $[8,9]$ and is realised by the map $x \mapsto x+3$. The *cut* is represented by the segment $[5,6]$. The execution of G and H, illustrated on the right-hand side of Figure 1, is composed of two paths: the restriction of the edge of G to the segment $[1,2]$, and the composition of the two edges.

Based on the notion of alternating cycle – defined easily from the notion of alternating paths, one defines a measurement $[\![\cdot,\cdot]\!]_m$ of couples of graphings and taking values in $\bar{\mathbf{R}}_+$. This measurement is parametrized[9] by the choice of a measurable map $m : \Omega \to \bar{\mathbf{R}}_+$. It is a quite involved work to define and study, and the results of this paper are based only on the existence of such a measurement and not its definition, so we refer the interested reader to our previous paper [38]. In the specific case of graphs – which are graphings over a discrete space – this measurement simply equals the sum, over the set of alternating cycles π, of $m(\omega(\pi))$ where m is any map $\Omega \to \bar{\mathbf{R}}_+$. This notion of measurement is extended to couples (a, A) where a is a real number (potentially infinite) and A a graphing; the consideration of this additional real number – the wager – finds its reasons in technical details that are explained in previous papers [33, 37]. The resulting couples, called projects, are used to interpret proofs.

Definition 5 (Project)**.** A project is a pair $\mathfrak{a} = (a, A)$ with $a \in \bar{\mathbf{R}}_+$ and A is a formal weighted sum of graphings $A = \sum_{i \in I^A} \alpha_i^A A_i$. We write $\mathbf{1}_A$ the sum $\sum_{i \in I^A} \alpha_i$.

From the measurement, one defines a notion of orthogonality that accounts for linear negation. This orthogonality relation is used to define *conducts*, specific sets of projects which will interpret formulas.

Definition 6 (Orthogonality)**.** Two projects $\mathfrak{a} = (a, A)$ and $\mathfrak{b} = (b, B)$ of equal supports are orthogonal, denoted $\mathfrak{a} \perp \mathfrak{b}$, when $a\mathbf{1}_B + b\mathbf{1}_A + [\![A, B]\!]_m \neq 0, \infty$.

Given a set T of projects, its orthogonal $T^\perp$ is defined as $\{\mathfrak{a} \mid \forall \mathfrak{b} \in T, \mathfrak{a} \perp \mathfrak{b}\}$. We will denote $T^{\perp\perp}$ the set $(T^\perp)^\perp$.

Definition 7 (Conduct)**.** A *conduct* $\mathbf{A}$ of support V is a set of projects of support V which is bi-orthogonally closed: $\mathbf{A} = \mathbf{A}^{\perp\perp}$.

Finally, one can define a category whose objects are conducts and morphisms are projects and which is shown to interpret multiplicative-additive linear logic. We do not detail this construction since it is quite involved. However, let us point out that the resulting model is completely non-degenerate (none of the connectives or constants are identified) and does not satisfy the mix and weakening rules [37].

Theorem 1 (Seiller [38])**.** *Let $\mathbf{X}$ be a measure space, $\mathfrak{m}$ a microcosm, Ω a monoid of weights. For all measurable map $m : \Omega \to \bar{\mathbf{R}}_+$, conducts and projects built from Ω-weighted $\mathfrak{m}$-graphings, with the orthogonality defined from the measurement defined from m, form a model of Multiplicative-Additive Linear Logic.*

3. The model

To describe the model, we will pick a measure space $\mathbf{X}$ together with a microcosm $\mathfrak{U}_\rho$ which are defined below. The construction we describe will not depend on the choices of Ω and $m : \Omega \to \mathbf{R}_{\geqslant 0}$, and therefore describes a family of quantitative models of second order linear logic.

Although the underlying space used here differs from our earlier work on exponentials [35], both are equivalent up to a Borel automorphism. The presentation we chose to work with here has the advantage of showing more explicitly the dynamics at work, while gaining intuitions from standard work on exponentials. Indeed, we chose to work with the Hilbert cube $[\mathbf{0},\mathbf{1}]^{\mathbf{N}}$, underlying an intu-

[9] We won't dwell on this choice of parameter in this paper, in order to avoid unnecessary complications. Although we here mention it for the sake of exactness, it will not play any specific role here. A fine analysis of the models would imply a consideration of specific values of m, but none of the results obtained in this paper depend on the choice of m.

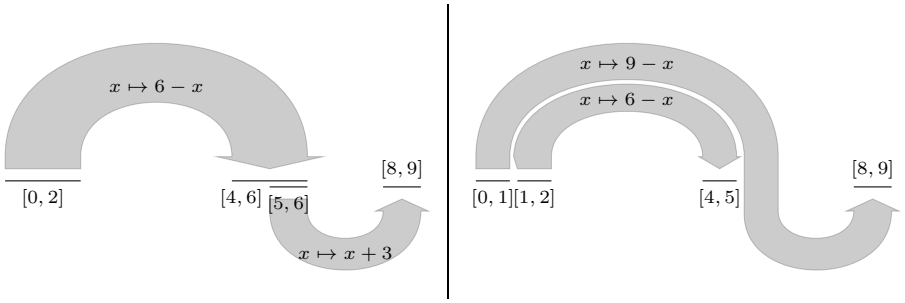


Figure 1: Example of an execution between two graphings.

itive correspondence[10] between *boxes* used to treat exponentials in proof nets and the copies of $[\mathbf{0}, \mathbf{1}]$.

Definition 8 (The space)**.** We define the measure space $\mathbf{X} = \mathbf{R} \times [\mathbf{0}, \mathbf{1}]^{\mathbf{N}}$, product of the real line with the Hilbert cube, endowed with its usual Borel algebra and Lebesgue measure.

Notations. We will write elements of $\mathbf{X}$ as couples (a, s), where $a \in \mathbf{R}$ and s is a sequence of elements in $[\mathbf{0}, \mathbf{1}]$. We will sometimes write sequences as $s \bullet s'$, i.e. as the concatenation of a finite sequence $s = (x_1, \ldots, x_k)$ and a sequence s'; when s contains only one element x we will identify x and (x). When considering elements of the space $\mathbf{X} \times [\mathbf{0}, \mathbf{1}]$, we will use a natural extension of this notation, and write them (a, s, e), with $(a, s) \in \mathbf{X}$ and $e \in [\mathbf{0}, \mathbf{1}]$.

We now define the microcosm, denoted $\mathfrak{U}_\rho$, that will be used to interpret proofs. We could very well have worked with the biggest microcosm possible (the so-called *macrocosm*) or any microcosm containing $\mathfrak{U}_\rho$. It is however more interesting to point out exactly the principles that are necessary to interpret second-order linear logic.

Definition 9 (The microcosm)**.** Let ρ be a measure-preserving bijection $[\mathbf{0}, \mathbf{1}]^2 \to [\mathbf{0}, \mathbf{1}]$. We define the microcosm $\mathfrak{U}_\rho$ as the monoid of measurable[11] maps $\mathbf{X} \times [\mathbf{0}, \mathbf{1}] \to \mathbf{X} \times [\mathbf{0}, \mathbf{1}]$ generated by:

- affine transformations on $\mathbf{R}$: $A^\alpha_\lambda : (x, s, e) \mapsto (\alpha x + \lambda, s, e)$;
- (finite) permutations on $[\mathbf{0}, \mathbf{1}]^{\mathbf{N}}$: $P_\sigma : (x, s, e) \mapsto (x, \sigma(s), e)$;
- the map $D_\rho : (a, (x, y) \bullet s, e) \mapsto (a, \rho(x, y) \bullet s, e)$ and its inverse;
- the *exchange* $\mathtt{xch}$: $(a, x \bullet s, e) \mapsto (a, e \bullet s, x)$

Notice that the exchange $\mathtt{xch}$ is an example of map that could not arise from a microcosm of maps from $\mathbf{X}$ to itself. This added principle is crucial for the definition of both dereliction and digging. Intuititvely, the microcosm of Definition 9 *without the exchange map* allows for Elementary Linear Logic[12], in the same spirit as our previous work on exponentials [35]; the added principle – the exchange – adds both dereliction and digging simultaneously.

Remark 1. One actually considers thick and sliced graphings up to a larger equivalence than that of variants. Indeed, the sliced and thick graphing $\sum_{i=1}^k \frac{1}{k} A_i$ is considered equivalent to the *universal*[13] graphing H whose restriction to the part of the dialect $[(i-1)/k, i/k]$ is equal to A_i, modulo the affine transformation $[(i-1)/k, i/k] \to [0, 1], x \mapsto (x \times k) - i + 1$.

By Theorem 1, we know that for any choices of Ω and m, the induced model interprets MALL. We will thus concentrate on exponential connectives here and refer the interested reader to earlier papers for the definition of MALL connectives.

4. The Exponentials

We now define the perennisation, that is the operation turning a project $\mathfrak{a}$ into a project $!\mathfrak{a}$ that can be duplicated. Indeed, in the interaction graphs models a project $\mathfrak{ctr}$ interpreting the contraction of arguments can be defined but it actually implements contraction, i.e. satifsfies $\mathfrak{ctr} :: \mathfrak{a} \equiv \mathfrak{a} \otimes \mathfrak{a}$ for a natural notion of equivalence $\equiv$, *only if* the graphing G is dialect-free [34, 35]. Thus the need for a *perennisation* operation that turns a project $\mathfrak{a}$ into a dialect-free project $!\mathfrak{a}$; this operation will in turn, when lifted to conducts, define the exponential connective.

In order to preserve all information contained in the dialect, the operation on graphings that underlies the perennisation will encode the information contained in the dialect in the support of the graphing. It is to be noted that the perennisation operation is not defined on all projects, but on the subset of so-called *balanced* projects. These are in particular projects whose dialect is equal to $[\mathbf{0}, \mathbf{1}]$ or, by extension, whose graphings are "balanced" sums: $\sum_{i=1}^k \alpha_i A_i$ such that $\alpha_i = 1/k$ for all i. This does not hinder the interpretation of proofs since projects arising from such an interpretation will all satisfy these conditions. Once this restriction is considered, the operation itself is easy to define: from a balanced project $(0, A)$ we construct $(0, !A)$ where $!A$ is the graphing obtained by "pushing" the dialect of A into the first slot of the sequence $[\mathbf{0}, \mathbf{1}]^{\mathbf{N}}$. For technical reasons explained later, we also need to create a fresh new copy of $[\mathbf{0}, \mathbf{1}]$ that will be used for implementing the promotion rule.

Definition 10. A project $\mathfrak{a} = (a, A)$ is *balanced* if $a = 0$ and the dialect of A is $[\mathbf{0}, \mathbf{1}]$. If E is a set of projects, we write $\mathrm{bal}(E)$ the subset of balanced projects in E.

[10] More precisely, the correspondence would be between boxes and copies of $[\mathbf{0}, \mathbf{1}] \times [\mathbf{0}, \mathbf{1}]$, cf. the definition of exponential connectives.

[11] We notice that those are all Borel automorphisms, thus in particular Borel-preserving and non-singular.

[12] To be more exact, the microcosm allowing for a model of ELL is the microcosm $\mathfrak{U}_\rho$ without the exchange but with the maps D_σ which permute the family of intervals $\{[(i-1)/k, i/k]\}_{i=1}^k$ in the dialect along a permutation σ of $\{1, \ldots, k\}$. Without these maps, one cannot define contraction as one cannot represent *slice-changing edges* [35]; it is not necessary to have all of them, though, as for instance all such D_σ for permutations σ over sets $\{1, \ldots, 2^p\}$ are enough. Notice that these maps – in the case $k = 2^p$ – are elements of $\mathfrak{U}_\rho$, defined as $D_\sigma = \mathtt{xch} \circ \rho_{(p)} \circ P_\sigma \circ \rho_{(p)}^{-1} \circ \mathtt{xch}$, where $\rho_{(p)}$ is recursively defined by:

$$\rho_{(0)} = \rho \qquad \rho_{(p)} = \rho_{(p-1)} \circ (\textstyle\prod_{i=1}^{2^{p-1}} \rho)$$

[13] This is the *smallest* such graphing, i.e. if H' also satisfies this property, then H is included in H'.

In order to define exponentials, we will need the following map:

$$B : \left\{ \begin{array}{rcl} \mathbf{X} \times [\mathbf{0},\mathbf{1}] & \to & \mathbf{X} \\ (a,s,d) & \mapsto & (a, d \bullet s) \end{array} \right.$$

This map B will be our way of encoding the dialect of A in the support of the graphing $!A$. This way, the resulting graphing $!A$ will contain the exact same information as A, but will be dialect-free. Though it might seem a transparent and useless operation, the fact that the dialect is now part of the support makes the graphings $!A$ and A behave quite differently when put into interaction with other projects. Intuitively, while the dialect is something private – e.g. control states – the support is not, and some projects might interact with $!A$ non-uniformly w.r.t. the former dialect of A.

Example 2. We consider two graphings, say G and H, both though of[14] as graphings with dialects $\{1,2\}$. Now, suppose that G and H are of type $\mathbf{A}$ and $\mathbf{A} \multimap \mathbf{B}$ respectively. Their execution $F :: G$ is then of type $\mathbf{B}$, and its dialect should be thought of as $\{1,2\} \times \{1,2\}$. We can also consider $!G$ and $!H$, which are of respective types $!\mathbf{A}$ and $!(\mathbf{A} \multimap \mathbf{B})$, and their execution $!G :: !H$. Let us explain why the latter cannot be of type $!\mathbf{B}$. Figure **??** illustrates this situation with examples of graphings G, $!G$, H, $!H$, as well as lists of the edges (alternating paths) of $G :: H$ and $!G :: !H$.

The execution of $!G$ and $!H$ actually produces the graphing defined as follows: compute the execution of G and H *as if they did not have any dialect*, and then take the perennisation of the result. In other words, the only alternating paths computed between G and H are those where the states of G and H are equal: this creates new paths (pictured in red path in Figure **??**), this deletes paths (the blue paths), and leaves some of them "unchanged". As a consequence, we cannot prove that $!G :: !H$ is of type $!\mathbf{B}$ since the only graphing we know for sure to belong to this type is $!(G :: H)$.

Definition 11 (Perennisation)**.** Let $\mathfrak{a} = (0, A)$ be a balanced project. We define its *perennisation* $!\mathfrak{a} = (0, !A)$ by considering the dialect-free graphing $!A = B^2(A \times \mathrm{Id}_{[\mathbf{0},\mathbf{1}]})$.

Definition 12 (Exponentials)**.** Let $\mathbf{A}$ be a conduct. We define the perennial conduct $!\mathbf{A}$ as the bi-orthogonal closure $!\mathbf{A} = (\sharp\mathbf{A})^{\perp\perp}$ where $\sharp\mathbf{A}$ is the set

$$\sharp\mathbf{A} = \{!\mathfrak{a} \mid \mathfrak{a} \in \mathrm{bal}(\mathbf{A})\}$$

5. A Model of Full Linear Logic

We already know from previous work that the model just described is a model of multiplicative-additive linear logic with second-order quantification [38]. To ensure that we have a sound interpretation of exponential connectives, we will show that the following principles can be implemented:

- functorial promotion $(!A \otimes !(A \multimap B)) \multimap !B$;
- dereliction $!A \multimap A$;
- digging $!A \multimap !!A$.

The principle of *contraction* $!A \multimap !A \otimes !A$ does not appear in this list as it holds for every possible definition of perennisation[15]. Let us notice moreover that the principle of functorial promotion was already shown to hold in our earlier work on exponentials [35]. We will however use here a less involved method for defining exponentials and implementing functorial promotion. The principles at work are more or less the same as in our earlier work, but this new implementation – inspired from recent work on complexity [39] – offers a clearer picture.

[14] Recall that we are actually working with "variants" G^c and H^c whose dialect are $[\mathbf{0},\mathbf{1}]$.

[15] As explained in Footnote 12, the microcosm already contains all the needed maps to define contraction.

The change of perspective illustrated in Example 2 is at the heart of the question of implementing functorial promotion. We want to "simulate" the disjointness of dialects. This is done in two steps: first make the encodings (in $!G$ and $!H$) of the dialects of G and H disjoint, by linking $!G$ and $!H$ through the permutation exchanging the two first copies of $[0,1]$. This corresponds to encoding the dialect of one of the two graphings on the second copy of $[0,1]$ instead of the first. Then we compute the result of this execution, obtaining a graphing which is almost $!(G :: H)$ except for the fact that its dialect is encoded on the two first copies of $[0,1]$ and not only on the first. We then use a specific graphing that will use the map D_ρ to encode this dialect on the first copy only.

Theorem 2. *Functorial Promotion holds.*

Proof (Sketch). The proof is much simpler in this setting than in our previous work on exponentials [35]. The principle is however quite the same: we use a first map to ensure the disjointness of the two "public dialects", and then we use a second map that will merge both copies. I.e. we define the maps:

$$\begin{array}{rcccl} \mathrm{twist} & : & (\lambda,(x,\rho(y,z)) \bullet s) & \mapsto & (\lambda,(y,\rho(x,z)) \bullet s) \\ \mathrm{merge} & : & (\lambda,(x,\rho(y,z)) \bullet s) & \mapsto & (\lambda,(\rho(x,y),z) \bullet s) \end{array}$$

To prove the result, we exhibit a project $\mathfrak{prom}$ and show that $\mathfrak{prom} \in !\mathbf{A} \otimes !(\mathbf{A} \multimap \mathbf{B}) \multimap !\mathbf{B}$. For this, we show that for all $!\mathfrak{a} = (0, !A) \in !\mathbf{A}$ and $!\mathfrak{f} = (0, !F) \in !(\mathbf{A} \multimap \mathbf{B})$, we have

$$\mathfrak{prom} :: !\mathfrak{a} :: !\mathfrak{f} = (0, \mathrm{merge}(!A :: \mathrm{twist}(!F)))$$

Finally, one easily checks that $\mathrm{merge}(!A :: \mathrm{twist}(!F))$ is equal to $!D$ where D is a variant of $F :: A$. □

Both digging and dereliction will work based on the simple idea that a continuous dialect $[\mathbf{0},\mathbf{1}]$ can be exchanged with a copy of $[\mathbf{0},\mathbf{1}]$ appearing in the Hilbert cube. This is exactly the computational principle encapsulated in the exchange map $\mathtt{xch}$. This implies that the *potential infinite* of dialects – i.e. the fact that a dialect can be any finite set, without bounds on its cardinality – can be managed within the projects themselves, something that could not be done in earlier constructions.

Theorem 3. *Digging holds.*

Proof (Sketch). As for the proof of Theorem 2, we exhibit an element $\mathfrak{dig}_{\mathbf{A}} \in !\mathbf{A} \multimap !!\mathbf{A}$. We show that, for all $!\mathfrak{a} = (0, !A)$ in $!\mathbf{A}$, one can compute $\mathfrak{dig}_{\mathbf{A}} :: !\mathfrak{a} = (0, \mathrm{push}(!A))$, where:

$$\mathrm{push} : (\lambda,(x,\rho(y,\rho(z,w))) \bullet s, e) \mapsto (\lambda;(e,z,x,y) \bullet s, w)$$

It is clear that $\mathrm{push}(!A)$ is equal to $!!A$, since the dialect of $!A$ (although $!A$ is dialect-free, not all elements of $!\mathbf{A}$ are, and therefore this is important) is encoded in the first copy of $[\mathbf{0},\mathbf{1}]$, while the second copy remains unused (this is because the second copy of $[\mathbf{0},\mathbf{1}]$ in $!A$ is unused[16]). □

The dereliction consists in "reconstructing" a dialect from a banged project. This can be performed using the same kind of tricks, i.e. using a continuous dialect.

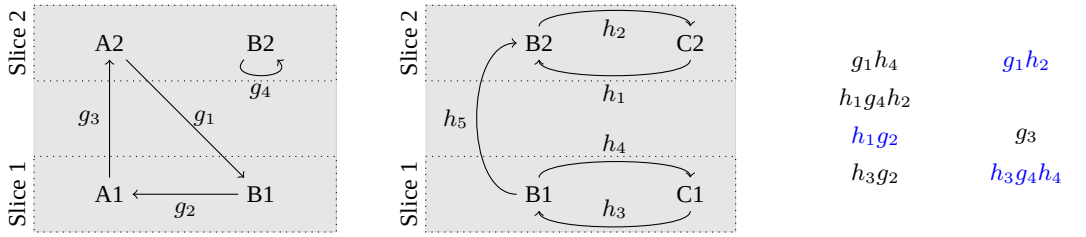
Theorem 4. *Dereliction holds.*

Proof (Sketch). Again, we exhibit an element $\mathfrak{der}_{\mathbf{A}} \in !\mathbf{A} \multimap \mathbf{A}$. For this, we show that for all $!\mathfrak{a} \in !\mathbf{A}$, one can compute $\mathfrak{der}_{\mathbf{A}} :: !\mathfrak{a} = (0, \mathrm{raise}(!A))$ where

$$\mathrm{raise} : (\lambda,(x,y) \bullet s, e) \mapsto (\lambda, s, \rho(x,\rho(y,e)))$$

Again, one easily checks that $\mathrm{raise}(!A)$ is equal to A. □

[16] As this is not the case for elements of $\mathbf{A}$, this explains why digging is not a co-dereliction.



(a) Graphing G (b) Graphing H (c) Edges of $G :: H$

(d) Graphing $!G$ (e) Graphing $!H$ (f) Edges of $!G :: !H$

Figure 2: Illustration of Example 2 with graphing seens as graphs.

Notice that the maps used to interpret dereliction and digging are not the only ones that satisfy the right properties, e.g. if one replaces raise by the map $\text{raise}^{(2)}(\lambda,(x,y)\bullet s,e) \mapsto (\lambda,s,\rho(\rho(x,y),e))$, we still have $\text{raise}^{(2)}$. The exact expressions are however important when to ensure that the execution soundly represents cut-elimination.

6. Interpretation of proofs

We first recall the notion of winning projects [35]. Winning projects are the equivalent of game semantics' winning strategies or classical realisability's proof-like terms. In particular, all interpretations of proofs will be winning projects.

Definition 13. A project $\mathfrak{a} = (a, A)$ is *winning* if it is balanced and if A is a disjoint union of transpositions, i.e. each edge e in A has a reverse edge e^* with $\phi^A_{e^*} = (\phi^A_e)^{-1}$ and the sources of edges are pairwise disjoint.

We now recall the basics of the proof system for which we define the interpretation of proofs. We are working with three different kinds of formulas, *positive*, *negative* and *neutral*. The technical reasons behind this are explained in our work on ELL [35]. Intuitively, neutral formulas correspond to the fragment of linear logic which does not allow for structural rules, negative formulas are those created from a perennisation while positive formulas are duals of negative formulas. They are defined inductively through the grammar shown on Figure 3 (neutral formulas are denoted by B which stands for *behavior* [35]).

Definition 14. A sequent $\Delta \Vdash \Gamma;\Theta$ is such that Δ,Θ contain only negative formulas, Θ containing at most one formula and Γ containing only neutrals.

Definition 15 (The System LL_{pol})**.** A proof in the system LL_{pol} is a derivation tree constructed from the derivation rules of ELL_{pol} [35], which are nothing more than polarised variants of elementary linear logic sequent calculus rules – presented with functorial promotion, extended with the rules in Figure 4.

One can then extend the inductive interpretation of proofs defined for ELL_{pol} in earlier work [35] by interpreting the additional rules as follows: the interpretation $\|\pi\|$ of a proof π obtained from a proof π' by using a dereliction rule (resp. a digging rule) on $!\mathbf{A}$ is defined as the execution of $\|\pi'\|$ with the project $\mathfrak{der}_{\mathbf{A}}$ (resp. $\mathfrak{dig}_{\mathbf{A}}$).

Theorem 5. *For every proof π of a sequent $\Delta \Vdash \Gamma;\Theta$ in LL_{pol}, the interpretation $\|\pi\|$ is a winning project in $\|\Delta \Vdash \Gamma;\Theta\|$*

The proof of this result is uninteresting in itself and follows exactly the proof of the same result for the restricted system ELL_{pol} [34, 35]. The additional cases of dereliction and digging rules are completely transparent since the projects exhibited in the proofs of Theorems 3 and 4 are clearly winning projects.

Notice that it is an open question whether the exponential isomorphism between $!(A \,\&\, B)$ and $!A \otimes !B$ did hold in the ELL model[17]. In the model we just described, however, this isomorphism holds; one just has to write down the usual derivation which can be interpreted soundly in the model.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Vdash A, A^{\perp};}{!A \Vdash A;}\,\text{der}}{!A, !B \Vdash A;}\,\text{weak} \quad \dfrac{\dfrac{\Vdash B, B^{\perp};}{!B \Vdash B;}\,\text{der}}{!A, !B \Vdash B;}\,\text{weak}}{!A, !B \Vdash A \,\&\, B;}\,\&}{!!A, !!B \Vdash; !(A \,\&\, B)}\,!}{!A, !!B \Vdash; !(A \,\&\, B)}\,\text{dig}}{!A, !B \Vdash; !(A \,\&\, B)}\,\text{dig}}{\Vdash; (!A \otimes !B) \multimap !(A \,\&\, B)}$$

This implies not only that that two conducts $!A \otimes !B$ and $!(A \,\&\, B)$ are isomorphic, but that they are equal. Indeed, the inclusion $!(A \,\&\, B) \subseteq !A \otimes !B$ can be proved as in our earlier paper [35]. Moreover, the interpretation $\|\pi\|$ of the above derivation can be shown to satisfy $\|\pi\| ::(!\mathfrak{a} \otimes !\mathfrak{b}) = !(\mathfrak{a} \,\&\, \mathfrak{b})$, where $\mathfrak{a} \,\&\, \mathfrak{b}$ is the usual construction of the $\&$ rule between $\mathfrak{a}$ and $\mathfrak{b}$, yielding the converse inclusion.

Theorem 6. *For any conducts* $\mathbf{A}$ *and* $\mathbf{B}$, $!(\mathbf{A} \,\&\, \mathbf{B}) = !\mathbf{A} \otimes !\mathbf{B}$.

[17] This is discussed in our earlier paper [35], but can be understood as follows: in the non-affine sequent calculus for ELL (presented with the functorial promotion rule) one cannot prove the implication $(!A \otimes !B) \multimap !(A \,\&\, B)$.

$$
\begin{array}{rcl}
B & := & X \mid X^{\perp} \mid \mathbf{0} \mid \mathbf{T} \mid B \otimes B \mid B \parr B \mid B \oplus B \mid B \,\&\, B \mid \forall X\, B \mid \exists X\, B \mid B \otimes N \mid B \parr P \\
N & := & \mathbf{1} \mid !B \mid !N \mid N \otimes N \mid N \,\&\, N \mid N \oplus N \mid N \parr P \mid \forall X\, N \mid \exists X\, N \\
P & := & \bot \mid ?B \mid ?P \mid P \parr P \mid P \,\&\, P \mid P \oplus P \mid N \otimes P \mid \forall X\, P \mid \exists X\, P
\end{array}
$$

Figure 3: Grammar for the formulas of $\mathrm{LL}_{\mathrm{pol}}$ (the symbol X denotes variables)

$$
\frac{\Delta, N \Vdash \Gamma; \Theta}{\Delta, !N \Vdash \Gamma; \Theta}\ \mathrm{der}^{pol} \qquad \frac{\Delta \Vdash \Gamma, B; \Theta}{\Delta, !B^{\perp} \Vdash \Gamma; \Theta}\ \mathrm{der} \qquad \frac{\Delta, !!N \Vdash \Gamma; \Theta}{\Delta, !N \Vdash \Gamma; \Theta}\ \mathrm{dig}
$$

Figure 4: Additional Rules for Dereliction and Digging

Before discussing the interpretation of cut-elimination in the models, we should add a few explanations about the soundness results we just obtained. Indeed, it should be stressed that the interpretation of proofs we just defined are non-trivial, i.e. that they do not identify (too much) distinct proofs. In the case of our models this is a consequence of the fact that two proofs have the same interpretation if and only if they have the same proof net [5, 14]. In other words, sequent calculus proofs are quotiented w.r.t. some (computationally inessential) commutations of rules.

What about cut-elimination? It is known that GoI does not represent cut-elimination exactly, i.e. it is not always the case that if π' is the normal form of π, then $\|\pi'\| = \mathrm{Ex}(\|\pi\|)$. It was shown that cut-elimination for MLL is soundly represented by execution [33], but there is a mismatch even in the exponential-free fragment because of additive cuts. This issue is discussed in details in previous work [37], where we solve this problem by considering a notion of *observational equivalence* $\cong$. Indeed, we showed that, even if $\|\pi'\| \neq \mathrm{Ex}(\|\pi\|)$ in presence of an additive cut, $\|\pi'\|$ and $\mathrm{Ex}(\|\pi\|)$ are observationally equivalent.

Theorem 7 (Seiller [37])**.** *If π' is obtained from π by applying a step of cut-elimination ($\&/\oplus$) then $\|\pi\| \cong \|\pi'\|$.*

We now consider the exponential connectives. We will consider a promotion rule cut against the following rules: dereliction, digging, and contraction. We consider for this a proof π of $\Vdash A^{\perp}, B$, or a proof π of $A \Vdash; B$ as both promotion rules are treated similarly, and its interpretation $\|\pi\| \in \mathbf{A} \multimap \mathbf{B}$. Applying a promotion rule (polarised or not) to π yields a proof ρ whose interpretation is $\|\rho\| = !\|\pi\| :: \mathfrak{prom}$. Then, given a proof π' to which we apply one of the three structural rules above to obtain a proof ρ', we consider the interpretations of the proof ν obtained by a cut between π and ρ, and the interpretation of the proof ν' obtained by applying a step of the cut-elimination procedure on ν. It turns out that those are equal, i.e. $\|\nu\| = \|\nu'\|$.

Theorem 8. *If π' is obtained from π by applying a step of cut-elimination among (promotion/dereliction), (promotion/digging) or (promotion/contraction), then $\|\pi\| = \|\pi'\|$.*

So execution computes these elimination steps on the nose. What about the last step, namely (promotion/weakening)? In that case, we are faced a problem similar to what happens for additive cuts in MALL. As execution is a completely local procedure, it cannot erase a whole proof at once. Thus, this elimination step is not soundly represented by the execution. However, one can show the following weaker result.

Theorem 9. *If π' is obtained from π by applying a step of cut-elimination among (promotion/weakening), then $\|\pi\| \cong \|\pi'\|$.*

However, these results we just showed are not enough to entail that cut-elimination is soundly represented by execution *up to observational equivalence*. In particular, one would need to show that perennisation and observational equivalence interact properly, i.e. one would hope for a result stating that, given two balanced projects $\mathfrak{a}$ and $\mathfrak{b}$, $\mathfrak{a} \cong \mathfrak{b}$ if and only if $!\mathfrak{a} \cong !\mathfrak{b}$. One can prove that $\mathfrak{a} \ncong \mathfrak{b}$ implies $!\mathfrak{a} \ncong !\mathfrak{b}$, however the converse implication is still an open question. We believe that this may be solved by considering a larger notion of equivalence, namely that of *balanced observational equivalence*, i.e. equivalence w.r.t testing by balanced projects.

7. Quantitative aspects of Interaction Graphs

One of the key aspects of interaction graphs is their ability to accomodate quantitative aspects. Formally, recent work by the author [36] relates the models (for multiplicative linear logic) obtained by the interaction graphs construction to so-called weighted relational models by Laird *et al.* [29]. It is thus natural to expect Interaction Graphs models of various classical quantitative models of computation such as probabilistic computation. It appears however that the construction is flexible enough to allow for the study of a much larger class of "quantitative aspects", as illustrated by two quite different examples that we now discuss: Kennedy's units of measure [23], and Ghica and Smith's bounded linear types [12].

We will limit the discussions of these examples here either to simply typed lambda-calculus, although the model is more expressive. Detailed study of the models in whole would lead us out of the scope of this paper, and should be the subject of future work.

7.1 Units of measure and bounded linear types

One of the most straightforward examples is the work by Kennedy [23] introducing types systems dealing with the notion of *units*. Here, the set of units one wants to deal with, e.g. meters, seconds squared, meters per second, is represented as an (abelian) group U. Then, types are considered dependent upon elements of this group, and one is allowed to consider the type of meters $\mathrm{Nat}[m]$ or the type of functions from masses to energies (such as multiplication by a squared velocity for instance). Moreover one is allowed to consider quantification over types of units. For instance, multiplication of natural numbers could be typed as $\forall u \forall u', \mathrm{Nat}[u] \times \mathrm{Nat}[u'] \Rightarrow \mathrm{Nat}[uu']$.

Now, if one considers the Interaction graph model described in this paper in which the monoid Ω is replaced by the chosen abelian group of units U, then we can easily interpret unit-dependent types and quantification over units. A very naive way to do so consists in appending to the interpretation of a term a single edge on a measurable subset used and reserved for the sole purpose of interpreting units, and weight this edge with the appropriate unit. For instance, a natural number of type $\mathrm{Nat}[1]$ (resp. of type $\mathrm{Nat}[m]$) will just be the usual interpretation of the natural number extended by a single edge of weight 1 (resp. of weight m). Moreover, polymorphism is here something we get for free, using the same methods as for second-

order quantification: the type $\forall u, \mathbf{A}[u]$ is simply interpreted as the conduct $\bigcap_{u\in U} \mathbf{A}[u]$.

Further to this limited setting of Kennedy's unit of measures type system, this same interpretation also provides models of bounded linear types. Bounded linear types were introduced by Ghica and Smith [12] as a generalisation of Girard, Scedrov and Scott's bounded linear logic BLL [21], a variation of linear logic that accounts for polynomial time computation. The definition of a bounded linear type system is dependent upon a resource semiring $(J, +, \times, 0, 1)$. It is clear that the underlying monoid $(J, \times, 1)$ could be chosen as the weight monoid in our models without hidering the naive interpretation considered above.

Thus, the only real challenge lies in the interpretation of the sum of the semiring, which is used only in the contraction rule. But one can choose to represent the sum through a splitting of the dialect $[0, 1]$. This leads to a correct interpretation of the (exponential) contraction rule since $\mathfrak{ctr}$ will map an element of the tensor in $\lambda.\mathbf{A} \otimes \mu.\mathbf{A}$ to an element of $(\lambda + \mu).\mathbf{A}$. In particular, it will take the two single edges encoding the elements of J and superimposing them using the dialect, creating the exact situation we chose for interpreting the sum.

Another way of thinking about this naive interpretation is that all that is needed to interpret these type systems is a single conduct $\mathbf{U}$ which possesses subconducts (i.e. subtypes) that can be used to simulate algebraic operations in the group U (for units of measure) or in the semiring J (for bounded linear types). Using notations from previous papers, the previous argumentation shows that this conduct $\mathbf{U}$ can be chosen as $!\mathbf{T}_{\mathbf{V}}$ with V a non-negligible measurable set. The naive interpretation $\|A\|$ is then obtained by defining the interpretation of a type A to be its classic interpretation $\|A\|_c$ in a simple type system tensored with $\mathbf{U}$, i.e. $\|A\| = \|A\|_c \otimes \mathbf{U}$.

The generality of our models seems to accomodate for the more general framework allowing for resource variables and resource polymorphism considered by dal Lago and Hoffman [27].

7.2 Probabilistic computation

We now discuss probabilistic computation. We first show how to interpret a simple typed probabilistic lambda calculus without using the monoid of weights, but only the fact that our dialect can be continuous. The probabilistic lambda-calculus we consider [6] is defined by the following grammar:

$$t := x \mid \lambda x.t \mid tt \mid \sum_{i=1}^{k} \alpha_i t_i$$

where in the last expression the coefficients α_i are elements of $[0, 1]$ which add up to 1.

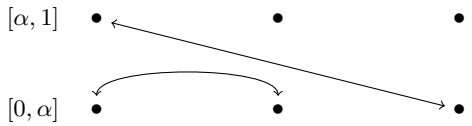
As mentioned in the introduction, future work will show how the models just defined can lead to interpretation of pure lambda-calculus. We are however bound to restrict to typable terms in this paper. We will consider simple types only, although nothing but space constraints prevent us for considering more expressive type system, e.g. system F. We thus consider simple types, and will use the following typing rule for the additional construct on terms (probabilistic sums):

$$\frac{\Gamma \vdash t_1 : P \qquad \dots \qquad \Gamma \vdash t_k : P}{\Gamma \vdash \sum_{i=1}^{k} \alpha_i t_i : P}$$

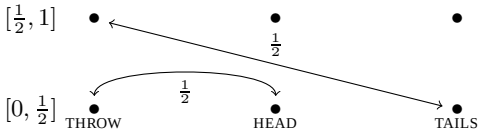
Now, we will interpret a given term of type A as an element of a conduct $!\mathbf{A}$. The use of the exponential modality will allow us to interpret the probabilistic sum using a variant of the contraction rule. In fact the interpretation can be understood as follows:

$$\frac{\dfrac{\Gamma \vdash t_1 : P \qquad \dots \qquad \Gamma \vdash t_k : P}{\Gamma \vdash <t_1, \dots, t_k> : \otimes_{i=1}^{k} P}}{\Gamma \vdash \sum_{i=1}^{k} \alpha_i t_i : P}$$

where the last operation is a *weighted contraction*. To illustrate how this weighted contraction works, we detail an example with a sum of two terms $\alpha t + (1 - \alpha t)$. The weighted contraction, instead of splitting the dialect into $[0, 1/2]$ on one hand and $[1/2, 1]$ on the other, will split it into $[0, \alpha]$ and $[\alpha, 1]$. This graphing can be understood as the realisation of the following thick graph[18]:



In order to really use the weights though, we will consider the example of a coin-toss operation, which then corresponds simply to the case $\alpha = 1/2$ in the above example. By considering the monoid of weights $[0, 1]$ with the usual multiplication, we can refine this interpretation by introducing non-trivial weights:



8. Conclusion

Girard's so-called "GoI3" model [19] already provided an interpretation of full linear logic. However, we managed to do so in a quantitative-flavoured framework. As examples, we explained how the models constructed can model various quantitative aspects, such as probabilistic computation, as well as some seemingly type-theoretic constructs such as modalities *à la* bounded linear logic, or Kennedy's units of measure. These exemplify the wide range of quantitative informations that can be dealt with in the models.

Beyond the results presented here, the adaptation of the interaction graphs framework to deal with continuous dialects results in a more mature and complete construction that opens new directions for future work. In particular, the possibility to restrict or expand the ways map can act on the dialect through the microcosm can be of interest in terms of computational complexity. Indeed, while the weight monoid and the measurement of weights seem to be related to different computational paradigms and can be used, for instance, for representing probabilistic computation, the microcosm can be used to restrict the computational principles allowed in the model and characterise in this way various complexity classes [39]. All characterisations considered in the cited work were based on variants of exponential connectives satisfying at least the contraction principle. In the more general construction explained here, we are now able to consider models of exponentials that do not satisfy this principle. In this line of work, it would be interesting to understand if one can adapt Mazza and Terui's work on parsimonious lambda-calculus [30–32], and obtain an interaction graph model for it.

Acknowledgments

This work was partly supported by the Marie Skłodowska-Curie Individual Fellowship (H2020-MSCA-IF-2014) 659920 - ReACT and the ANR 12 JS02 006 01 project COQUAS.

[18] Elements of the dialect (resp. support) are listed on a vertical (resp. horizontal) scale; double arrows represent two inverse edges.

References

[1] S. Adams. Trees and amenable equivalence relations. *Ergodic Theory and Dynamical Systems*, 10:1–14, 1990.

[2] V. Danos. *La Logique Linéaire Appliquée à l'Étude de Divers Processus de Normalisation (principalement du λ-calcul)*. PhD thesis, Paris VII University, 1990.

[3] V. Danos and T. Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Information and Computation*, 209, 2011.

[4] V. Danos and J.-B. Joinet. Linear logic & elementary time. *Information and Computation*, 183(1):123–137, 2003.

[5] V. Danos and L. Regnier. Proof-nets and the hilbert space. In *Advances in Linear Logic*, pp. 307–328. Cambridge University Press, 1995.

[6] A. Di Pierro, C. Hankin, and H. Wiklicky. Probabilistic λ-calculus and quantitative program analysis. *Journal of Logic and Computation*, 15 (2):159–179, 2005.

[7] T. Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15(4):615–646, 2005.

[8] D. Gaboriau. Coût des relations d'équivalence et des groupes. *Inventiones Mathematicae*, 139:41–98, 2000.

[9] D. R. Ghica. Geometry of synthesis: a structured approach to vlsi design. In M. Hofmann and M. Felleisen, editors, *POPL*, pp. 363–375. ACM, 2007.

[10] D. R. Ghica and A. Smith. Geometry of synthesis II: From games to delay-insensitive circuits. *Electr. Notes Theor. Comput. Sci.*, 265: 301–324, 2010.

[11] D. R. Ghica and A. Smith. Geometry of synthesis III: resource management through type inference. In T. Ball and M. Sagiv, editors, *Proc. of POPL*, pp. 345–356. ACM, 2011.

[12] D. R. Ghica and A. I. Smith. Bounded linear types in a resource semiring. In *Programming Languages and Systems*, *Lecture Notes in Computer Science* vol. 8410, pp. 331–350. Springer, 2014.

[13] D. R. Ghica, A. Smith, and S. Singh. Geometry of synthesis IV: compiling affine recursion into static hardware. In M. M. T. Chakravarty, Z. Hu, and O. Danvy, editors, *ICFP*, pp. 221–233. ACM, 2011.

[14] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[15] J.-Y. Girard. Normal functors, power series and λ-calculus. *Annals of Pure and Applied Logic*, 37(2):129–177, 1988.

[16] J.-Y. Girard. Geometry of interaction I: Interpretation of system F. In *In Proc. Logic Colloquium 88*, 1989.

[17] J.-Y. Girard. Towards a geometry of interaction. In *Proc. of the AMS Conference on Categories, Logic and Computer Science*, 1989.

[18] J.-Y. Girard. Light linear logic. In *Selected Papers from the International Workshop on Logical and Computational Complexity*, LCC '94, pp. 145–176, London, UK, UK, 1995. Springer-Verlag.

[19] J.-Y. Girard. Geometry of interaction III: Accommodating the additives. In *Advances in Linear Logic*, number 222 in Lecture Notes Series, pp. 329–389. Cambridge University Press, 1995.

[20] J.-Y. Girard. Geometry of interaction V: Logic in the hyperfinite factor. *Theoretical Computer Science*, 412:1860–1883, 2011.

[21] J.-Y. Girard, A. Scedrov, and P. J. Scott. Bounded linear logic: a modular approach to polynomial-time computability. *Theor. Comput. Sci.*, 97(1):1–66, Apr. 1992.

[22] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285–408, 2000.

[23] A. J. Kennedy. Relational parametricity and units of measure. In *Proc. of POPL*, pp. 442–455. ACM, 1997.

[24] S. C. Kleene. On the interpretation of intuitionistic number theory. *Journal of Symbolic Logic*, 10, 1945.

[25] J.-L. Krivine. Typed lambda-calculus in classical zermelo-fraenkel set theory. *Archive for Mathematical Logic*, 40(3):189–205, 2001.

[26] J.-L. Krivine. Realizability in classical logic. *Panoramas et synthèses*, 27:197–229, 2009.

[27] U. d. Lago and M. Hofmann. Realizability models and implicit complexity. *Theoretical Computer Science*, 412:2029–2047, 2011.

[28] U. d. Lago and O. Laurent. Quantitative game semantics for linear logic. In *Proc. of CSL*, *Lecture Notes in Computer Science* vol. 5213, pp. 230–245. Springer, 2008.

[29] J. Laird, G. McCusker, G. Manzonetto, and M. Pagani. Weighted relational models of typed lambda-calculi. In *Proc. of LICS*, 2013.

[30] D. Mazza. Non-uniform polytime computation in the infinitary affine lambda-calculus. In *Proc. of ICALP*, *Lecture Notes in Computer Science* vol. 8573, pp. 305–317. Springer, 2014.

[31] D. Mazza. Simple parsimonious types and logarithmic space. In *Proc. of CSL*, pp. 24–40, 2015.

[32] D. Mazza and K. Terui. Parsimonious types and non-uniform computation. In *Proc. of ICALP, Part II*, *Lecture Notes in Computer Science* vol. 9135, pp. 350–361. Springer, 2015.

[33] T. Seiller. Interaction graphs: Multiplicatives. *Annals of Pure and Applied Logic*, 163:1808–1837, December 2012.

[34] T. Seiller. *Logique dans le facteur hyperfini : géometrie de l'interaction et complexité*. PhD thesis, Aix-Marseille Univ., 2012.

[35] T. Seiller. Interaction graphs: Exponentials. *Logical Methods in Computer Science*. Submitted, 2014. Under revision.

[36] T. Seiller. From Dynamic to Static Semantics, Quantitatively. Submitted, 2016.

[37] T. Seiller. Interaction graphs: Additives. *Annals of Pure and Applied Logic*, 167:95 – 154, 2016.

[38] T. Seiller. Interaction graphs: Graphings. *Annals of Pure and Applied Logic*, 2016. To appear.

[39] T. Seiller. Towards a *Complexity-through-Realizability* theory. Submitted, 2016.